

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, информатики и информационных технологий  
Кафедра информатики, информационных технологий  
и методики обучения информатике

# ГЕНЕРАЦИЯ QR-КОДОВ

*Выпускная квалификационная работа  
бакалавра по направлению подготовки*

*02.03.02 – Фундаментальная информатика и информационные технологии*

Исполнитель: студент группы Б-41  
Носов В. В.

Руководитель: к.п.н., доцент кафедры  
ИИТиМОИ Газейкина А. И.

Работа допущена к защите  
«\_\_\_\_» \_\_\_\_\_ 2016 г.  
Зав. кафедрой \_\_\_\_\_

Екатеринбург – 2016

## **Реферат**

Носов В.В. Генерация QR-кодов, выпускная квалификационная работа: 42 стр., рис. 18, табл. 4, библи. 20 назв., приложений 0. Ключевые слова: QR-КОД, ГЕНЕРАЦИЯ QR-КОДА, АЛГОРИТМЫ ГЕНЕРАЦИИ QR-КОДА. Объект исследования – алгоритмы генерации и распознавания QR-кода. Цель работы – разработать приложение, реализующее генерацию QR-кода.

В работе описаны структурное описание QR-кода и алгоритмы его кодирования и различные динамические библиотеки и web-сервисы, предназначенные для генерации QR-кодов. Так же в работе описывается результат разработки приложения для генерации QR-кодов.

Система успешно прошла апробацию в Уральском государственном педагогическом университете. После некоторой адаптации разработанное приложение может быть использовано в различных сферах деятельности.

## Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ПРИЛОЖЕНИЙ, ИСПОЛЬЗУЮЩИХ QR-КОДЫ.....</b>	<b>6</b>
1.1    АНАЛИЗ СТРУКТУРЫ QR-КОДА.....	6
1.2    АНАЛИЗ ДИНАМИЧЕСКИХ БИБЛИОТЕК И WEB-СЕРВИСОВ ДЛЯ ГЕНЕРАЦИИ QR- КОДОВ            14	
1.3    ТЕХНИЧЕСКОЕ ЗАДАНИЕ .....	25
<b>ГЛАВА 2. РАЗРАБОТКА ПРИЛОЖЕНИЯ, ГЕНЕРИРУЮЩЕГО QR- КОД.....</b>	<b>29</b>
2.1    ИНФОРМАЦИОННЫЕ И ФУНКЦИОНАЛЬНЫЕ МОДЕЛИ ПРИЛОЖЕНИЯ .....	29
2.2    ОПИСАНИЕ ПРИЛОЖЕНИЯ (РЕЗУЛЬТАТА РАЗРАБОТКИ) .....	33
2.3    АПРОБАЦИЯ ПРИЛОЖЕНИЯ .....	35
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>39</b>
<b>ЛИТЕРАТУРА .....</b>	<b>41</b>
<b>ПРИЛОЖЕНИЯ .....</b>	<b>43</b>
Приложение 1.    Листинг программы.....	43

## **Введение**

В современном информационном обществе с каждым днем стремительно увеличивается уровень развития информационных и коммуникационных технологий. За последние годы их интенсивное использование и глобальное распространение, а также неограниченный доступ общества к интернету привели к многократному увеличению информации объема информации. В связи с этим возникает необходимость представления информации в компактном, простом в использовании, комфортном и визуально приятном для пользователя виде. Это поможет пользователю среди огромного объема сведений быстро и удобно найти необходимую информацию, затратив при этом минимум времени и усилий.

Жизнь современного человека уже нельзя представить без различных устройств, таких как мобильный телефон или планшетный компьютер, они стали неотъемлемым атрибутом каждого. Поэтому возникла необходимость в новой форме представления информации, которая будет отвечать обширному списку требований современного информационного общества. Так возникли QR-коды или «quick response», что переводится как быстрый отклик. Такие матричные коды представляют из себя двумерные штрих-коды, расшифровка которых проводится в двух измерениях, по горизонтали и по вертикали, что позволяет «закодировать» большой объем информации.

В настоящее время это один из самых популярных инструментов мобильной коммерции. Первоначально QR-коды использовались только в промышленности, на сегодняшний день они активно применяются в потребительской среде (реклама, онлайн покупки), в финансовой сфере и экономике (банковские терминалы), в сфере авиа и железнодорожных перевозок (информация с билетов), в образовании (формирование расписания), в культурной сфере (музеи), а также в медицине.

Таким образом, QR-код позволяет любому заинтересованному человеку, к примеру, мгновенно зайти на сайт, посвященный компании, продукту, историческому объекту и получить исчерпывающие данные.

В связи с этим можно утверждать, что создание программного приложения, которое будет зашифровывать информацию и генерировать QR-код на сегодняшний день является актуальной темой. Приложение, разработанное в ходе исследования, позволит, например, простым пользователям не вводить длинные ссылки на требуемый сайт вручную, а просто отсканировав изображение, автоматически попасть на web-страницу, то есть сократить объем выполненных пользователем действий, облегчить его труд и минимизировать потраченное время, не требуя от самого пользователя специальных знаний или навыков.

**Объект исследования:** QR-код.

**Предмет исследования:** Библиотеки и алгоритмы генерации QR-кода.

**Цель:** разработать приложение, реализующее генерацию QR-кода.

Для реализации цели были поставлены следующие **задачи**:

1. проанализировать научную литературу в области QR-кодов с целью изучения структуры QR-кодов;
2. проанализировать динамические библиотеки и web-сервисы, генерирующие QR-коды, провести сравнительный анализ алгоритмов генерации QR-кодов;
3. составить техническое задание;
4. разработать приложение, реализующее генерацию QR-кода;
5. провести апробацию приложения и разработать программную документацию по его применению.

# **Глава 1. Теоретические основы разработки приложений, использующих QR-коды**

## **1.1 Анализ структуры QR-кода**

Данные в QR-кодах закодированы с помощью черных и белых модулей, которые в дальнейшем могут быть расшифрованы специальными автоматизированными средствами за очень короткое время. При этом используя данную технологию можно зашифровать информацию различных видов начиная с обычного текста и заканчивая изображениями, адресами web-сайтов, номерами телефона.

Структура QR-кода представляет собой матрицу, содержащую двоичную информацию. В QR-код закодировано много разной информации[12]. Помимо тех данных, что закодировал пользователь в коде содержатся:

- версия кода;
- код маски;
- уровень коррекции ошибок.

Так же на QR-коде есть необходимые поля, они не несут закодированной информации, а содержат информацию о версии кода, о типе его кодирования и дополнительную информацию для его декодирования [16, с.51]:

- поисковые узоры;
- выравнивающие узоры;
- полосы синхронизации;
- код маски и уровня коррекции;
- код версии (начиная с 7 версии);
- тихая зона.



Рисунок 1. Структура QR-кода

Тихая зона представляет собой рамку из пустых модулей. Ширина такой рамки составляет 4 модуля.

Поисковые узоры расположены в углах QR-кода, кроме правого нижнего, и представляют собой три мишени, по которым определяется ориентация и расположение QR-кода. Поисковые узоры состоят из квадрата черных модулей размером 3x3 модуля, вокруг этого квадрата рамка из белых модулей, шириной 1 модуль и рамка из черных модулей, так же шириной 1 модуль. От остального кода мишень ограждается половиной рамки из белых модулей шириной в 1 модуль.

Выравнивающие узоры используются со второй версии кода для более точного считывания кода. Такие узоры размещаются на разных позициях, в зависимости от версии кода. Выравнивающие узоры не могут накладываться на поисковые узоры. Выравнивающие узоры представляют собой рамку из черных модулей размером 5x5 и шириной 1 модуль, в центре узора находится 1 черный модуль, окруженный рамкой из белых модулей.

Полосы синхронизации используются для определения размера модулей. Полосы располагаются по горизонтали от верхнего левого поискового узора (от нижнего правого угла) до верхнего правого поискового узора (до нижнего

левого угла), и вертикальная полоса идет от левого верхнего поискового узора (от нижнего правого угла) до левого нижнего узора (до верхнего правого угла). Полосы идут непосредственно от поискового узора, минуя ограничительную белую рамку. Полосы синхронизации не могут накладываться на выравнивающие узоры. Полосы представляют собой чередование белых и черных модулей.

Код маски и уровня коррекции расположен рядом с поисковыми узорами:

- под правым верхним (8 модулей);
- справа от левого нижнего (7 модулей);
- дублируются по бокам левого верхнего (8 модулей), с пробелом на 7 модуле (там, где проходят полосы синхронизации).

Код версии используется считывающим устройством для определения версии кода[16, с.70]. Находится слева от верхнего правого поискового узора и дублируется сверху от нижнего левого поискового узора. Дублирование происходит по правилу: код версии у верхнего правого поискового узора отображают зеркально и поворачивают на 90 градусов против часовой стрелки.

В отличие от старого одномерного штрихкода, который считывается механически, с помощью маленького пучка света, QR-код считывается с помощью двухмерного цифрового датчика изображений (фотокамерой телефона, web-камерой), а затем в цифровой форме анализируется программируемым процессором. Процессор находит три поисковых узора в углах цифрового образа QR-кода и нормализует изображение. Затем считываются модули кода и преобразуются в двоичные числа.

В QR-код можно закодировать самую разную информацию, и для этого он должен поддерживать различные типы кодирования, например [13]:

- числовое;
- символично-числовое;
- байтовое;



- кандзи.

### Числовое кодирование

Кодирование в этом режиме происходит посредством разбиения всей последовательности символов на равные части по 3 символа, после чего, каждая группа переводится в двоичное число и записывается в 10 бит информации, после чего добавляется к общей последовательности бит [16, с.19]. Если количество символов в кодируемом числе не кратно 3, то если в конце остается 1 символ, то его кодируют 4 битами, а если осталось 2 символа, то его кодируют 7 битами.

Таким образом, число «1506» преобразуется в строку: «00100101100000000110».

$$\begin{aligned}
 1506 &= 150 + 6 & 1506_{10} &= 0010010110 + 00000110 \\
 150_{10} &= 0010010110 & 1506_{10} &= 001001011000000110 \\
 6_{10} &= 00000110
 \end{aligned}$$

Рисунок 2. Числовое кодирование

### Буквенно-числовое кодирование

В отличие от числового кодирования, для символьного кодирования последовательность символов разбивается по 2 символа и кодируется в 11 бит информации. В каждой группе каждый символ кодируется согласно таблице «Значения символов в буквенно-цифровом кодировании» [рис.2][19]. Произведение значения первого символа на 45 складывается со значением второго символа, после чего записывается в 11-битное двоичное число и добавляется к общей битовой строке. Если в конце строки остается один символ, то его значение кодируется 6 битами. Таким образом, строка «source» будет записана как «101000001001010110000101000101010»

$source = so + ur + se$   
 $so = 28 * 45 + 24 = 1260 + 24 = 1284 = 10100000100$   
 $ur = 30 * 45 + 27 = 1350 + 27 = 1377 = 10101100001$   
 $se = 28 * 45 + 14 = 1260 + 14 = 1274 = 10011111010$   
 $source = 10100000100 + 10101100001 + 10011111010$   
 $source = 101000001001010110000110011111010$

Рисунок 3. Буквенно-числовое кодирование

0	0	A	10	K	20	U	30	+	40
1	1	B	11	L	21	V	31	-	41
2	2	C	12	M	22	W	32	.	42
3	3	D	13	N	23	X	33	/	43
4	4	E	14	O	24	Y	34	:	44
5	5	F	15	P	25	Z	35		
6	6	G	16	Q	26	Пробел	36		
7	7	H	17	R	27	\$	37		
8	8	I	18	S	28	%	38		
9	9	J	19	T	29	*	39		

Рисунок 4. Значения символов в буквенно-цифровом кодировании

### Байтовое кодирование

Байтовое кодирование – это универсальный тип кодирования, которым можно закодировать любую последовательность данных. В случае с байтовым кодированием, данные переводятся в двоичную систему в неизменном виде. При таком кодировании строка «УрГПУ» будет выглядеть следующим образом: «1000010001110001000000100000100111000001111110000100011».

### Кандзи кодирование

В основе кодирования Кандзи, как и в случае с буквенно-числовым кодированием, лежит визуально воспринимаемая таблица иероглифов с их кодами. Для японского языка основное значение имеют две таблицы символов:

- JIS 0208:1997;
- JIS 0212:1990.

## Версия кода

Коды делятся на версии от 1 до 40. Каждая версия имеет особенности в конфигурации и количестве модулей, составляющих QR-код. Размер 1 версии – 21х21, а версии 40 – 177х177. С каждой версией размерность кода увеличивается на 4 модуля.

Для каждой версии кода определена емкость с учетом уровня коррекции ошибок. Чем больше информации необходимо закодировать и чем больший уровень коррекции используется, тем большая версия кода потребуется. Большинство современных QR-генераторов автоматически подбирают версию кода с учетом уровня коррекции ошибок

## Код маски

Маска используется для повышения надежности сканирования QR-кода, посредством формирования такой матрицы кода, на которой было бы как можно меньше смежных модулей одинакового цвета. Для выбора маски формируются 8 матриц с одинаковым информационным содержанием. На каждой матрице отдельная маска, которая суммирует исходную матрицу QR-кода по правилу «исключающее ИЛИ» (xor). Полученные матрицы подвергаются математической обработке, где ведется подсчет «штрафного» коэффициента, после чего выбирают одну матрицу. Для выбора оптимальной матрицы значение «штрафного» коэффициента должно быть минимальным [20].

Ниже приведена таблица масок и соответствующих математических формул для подсчета «штрафного» коэффициента (Таб.1). В математической формуле маски используются номера строк (j) и номера столбцов (i). В каждой матрице рассматривается каждый информационный модуль. Если при применении формулы на модуле соотношение верно, то модуль инвертируется.

*Таблица 1.*

*Математические формулы масок QR-кода*

Маска QR-кода	Математическая формула маски
---------------	------------------------------

000	$(i + j) \bmod 2 = 0$
001	$i \bmod 2 = 0$
010	$j \bmod 3 = 0$
011	$(i + j) \bmod 3 = 0$
100	$((i \div 2) + (j \div 3)) \bmod 2 = 0$
101	$(i * j) \bmod 2 + (i j) \bmod 3 = 0$
110	$((i * j) \bmod 2 + (i j) \bmod 3) \bmod 2 = 0$
111	$((i + j) \bmod 2 + (i j) \bmod 3) \bmod 2 = 0$

#### Уровень коррекции ошибок

В структуре QR-кода так же присутствует механизм повышения надежности хранения информации, так называемый уровень коррекции ошибок. Уровень коррекции ошибок строится на основе алгоритма Рида-Соломона [16, с.31]. Существует 4 вида уровня коррекции, которые задаются при создании QR-кода. Уровень коррекции ошибок способствует увеличению надежности хранения информации, что позволяет при потере части кода восстановить зашифрованную информацию, например, самый высокий уровень коррекции ошибок (H), позволяет прочесть код с потерей 30%. Однако, уровень коррекции приводит к увеличению размера самого кода (Таб.3). Ниже приведена таблица уровней коррекции ошибок, на которой показаны допустимые повреждения для корректного считывания QR-кода (Таб.2).

*Таблица 2.*

*Допустимые повреждения для корректного считывания QR-кода.*

Уровень коррекции	Допустимый процент нарушений
L	до 7%
M	до 15%
Q	до 25%
H	до 30%

В таблице 3 приведены различные типы кодирования, версии кода, уровни коррекции ошибок и размеры QR-кода, а так же соотношения в виде количества символов, которое можно закодировать в QR-код [11]. Например в QR-код версии 3 при буквенно-числовом кодировании и уровнем коррекции ошибок «Q», можно закодировать 27 символов.

Таблица 3.

*Размеры QR-кода с учетом уровня коррекции ошибок и версии кода*

Версия QR-кода	Размер QR-кода (модуль)	Уровень коррекции ошибок	Максимальное количество символов с учетом коррекции ошибок			
			Числовое кодирование	Буквенно-числовое кодирование	Байтовое кодирование	Кандзи кодирование
1	21x21	L	41	25	17	10
		M	34	20	14	8
		Q	27	16	11	7
		H	17	10	7	4
2	25x25	L	77	47	32	20
		M	63	38	26	16
		Q	48	29	20	12
		H	34	20	14	8
3	29x29	L	127	77	53	32
		M	101	61	42	26
		Q	77	47	32	20
		H	58	35	24	15
4	33x33	L	187	114	78	48
		M	149	90	62	38
		Q	111	67	46	28
		H	82	50	34	21
10	57x57	L	652	395	271	167
		M	513	311	213	131
		Q	364	221	151	93

		H	288	174	119	74
40	177x177	L	7089	4296	2953	1817
		M	5596	3391	2331	1435
		Q	3993	2420	1663	1024
		H	3057	1852	1273	784

Таким образом, анализ структуры QR-кода показал, что QR-код – это удобное, эффективное средство для хранения и использования информации различных видов, которое обладает механизмом повышения надежности хранения информации.

## **1.2 Анализ динамических библиотек и web-сервисов для генерации QR-кодов**

Для генерации QR-кодов существует множество решений. Например для генерации можно использовать уже готовые web-сервисы, а можно использовать приложения, построенные на специальных динамических библиотеках, которые облегчают работу программиста и выполняют основные задачи по генерации QR-кода, в таком случае программисту нужно только правильно использовать динамические библиотеки с помощью специальных зарезервированных команд.

WEB-сервисы:

В настоящее время существует множество различных бесплатных online-сервисов, которые предоставляют возможность создания QR-кодов.

В процессе работы были протестированы следующие онлайн генераторы QR-кодов:

1. [qrcoder.ru](http://qrcoder.ru);
2. [qr-coder.ru](http://qr-coder.ru);
3. [qrcc.ru](http://qrcc.ru);
4. [qreambee.ru](http://qreambee.ru);
5. [qrcode.kaywa.com](http://qrcode.kaywa.com);

6. [qrcode.com.ua](http://qrcode.com.ua).

[Qrcoder.ru](http://Qrcoder.ru)

Данный сервис имеет простой интуитивно понятный интерфейс [7].

Средствами этого сервиса можно сгенерировать QR-код, содержащий следующую информацию:

- текст;
- адрес сайта;
- визитка;
- SMS-сообщение.

После генерации можно задать размер модуля в пикселях.

Рисунок 5. web-сервис [qrcoder.ru](http://qrcoder.ru)

[Qr-coder.ru](http://Qr-coder.ru)

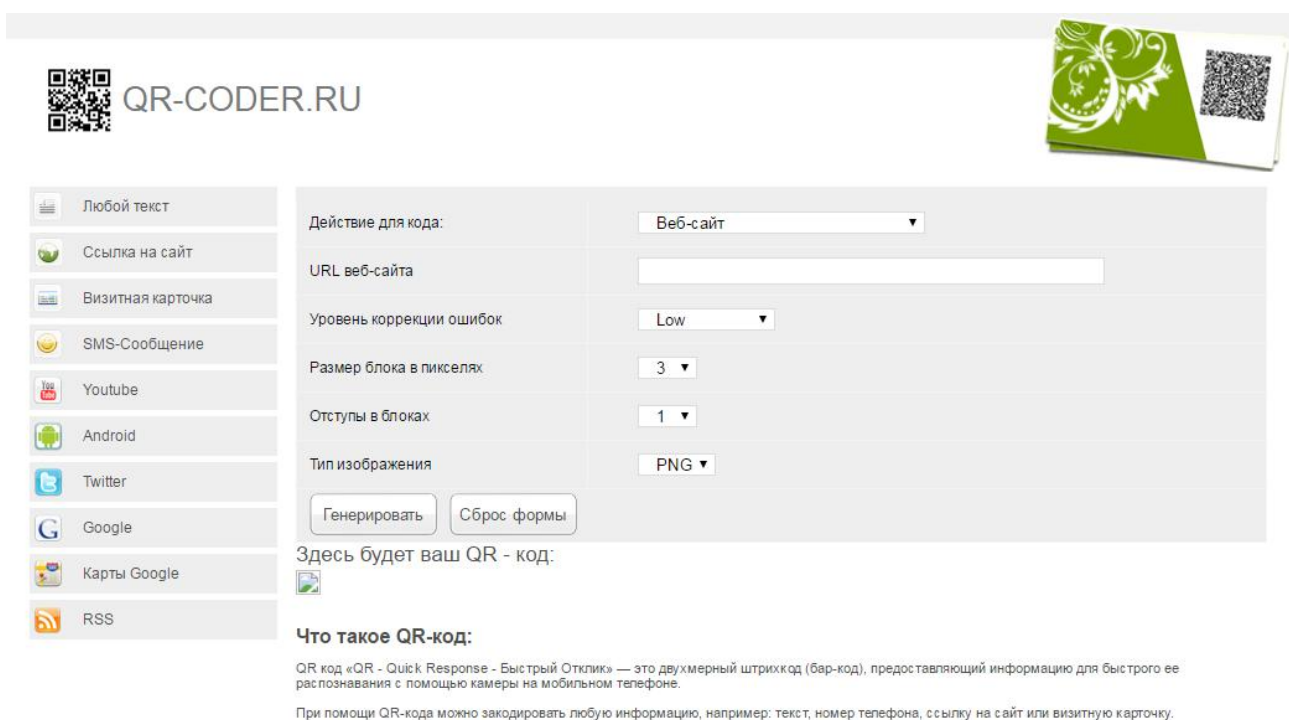
Средствами этого сервиса можно сгенерировать QR-код, содержащий следующую информацию [8]:

- текст;

- адрес сайта;
- визитка;
- SMS-сообщение;
- сообщение в Twitter;
- координаты Google Maps;
- ссылка на YouTube;
- поиск на Android Market.

После генерации QR-кода есть возможность настроить его, а именно:

- уровень коррекции;
- размер модуля в пикселях;
- расширение выходного файла.



QR-CODER.RU

Действие для кода: Веб-сайт

URL веб-сайта

Уровень коррекции ошибок: Low

Размер блока в пикселях: 3

Отступы в блоках: 1

Тип изображения: PNG

Генерировать Сброс формы

Здесь будет ваш QR - код:

**Что такое QR-код:**

QR код «QR - Quick Response - Быстрый Отклик» — это двухмерный штрихкод (бар-код), предоставляющий информацию для быстрого ее распознавания с помощью камеры на мобильном телефоне.

При помощи QR-кода можно закодировать любую информацию, например: текст, номер телефона, ссылку на сайт или визитную карточку.

Рисунок 6. web-сервис qr-coder.ru




Qrcc.ru

Интерфейс данного сервиса интуитивно понятен. С помощью этого сервиса можно сгенерировать QR-код, содержащий следующую информацию [4]:

- визитка;



- адрес сайта;
- текст;
- телефонный звонок;
- SMS-сообщение;
- координаты Google Maps;
- e-mail адрес;
- e-mail сообщение;
- запланированное событие;
- данные о сети Wi-Fi.

















QR-code
DataMatrix
Micro QR

Визитка (VCARD)	Имя	<input type="text"/>
Адрес сайта(URL)	Фамилия	<input type="text"/>
Произвольный Текст	Телефон домашний	<input type="text"/>
Телефонный звонок	Телефон рабочий	<input type="text"/>
CMS-сообщение	Телефон мобильный	<input type="text"/>
Координаты Google Maps	Факс	<input type="text"/>
E-mail адрес	E-mail	<input type="text"/>
E-mail сообщение	URL	<input type="text"/>
Запланированное событие(VCALENDAR)	Адрес	<input type="text"/>
Wi-Fi	Город	<input type="text"/>
	Страна	<input type="text"/>
	День рождения	Год <input type="text"/> Месяц <input type="text"/> День <input type="text"/>
	Организация	<input type="text"/>
	Должность	<input type="text"/>

Иконка/текст внутри кода:

НЕТ

Текст над кодом: 
Текст под кодом:

Цвет кода: #000000

☐

Цвет фона: #FFFFFF

☐

Цвет текста: #000000

☐

Размер кода:

x1

x2

x3

x4

Создать код

Рисунок 7. web-сервис qrcc.ru

Creambee.ru

Данный сервис имеет как бесплатные, так и платные функции. Платные функции предназначены для оптимизации маркетинговой политики различных предприятий. К дополнительным платным функциям относятся [9]:

- создание динамических QR-кодов;
- предоставление статистики сканирований.

Для простого пользователя будет достаточно бесплатных функций сервиса:

- простой текст;
- визитная карточка;
- номер телефона;
- SMS сообщение;
- ссылку на сайт;
- e-mail адрес;
- сообщение для Twitter;
- поделиться в Facebook.

После генерации кода имеется возможность выбора уровней коррекции. Так же можно изменять внешний вид кода, например добавлять или удалять сегменты кода ,что позволит придать ему желаемый внешний вид(изменения не касаются обязательных частей кода).

Что касается дизайнерских решений, то данный сервис располагает большим арсеналом настроек внешнего вида бар-кода;

- изменение цвета выбранных частей кода;
- трансформации и выбор формы отдельных модулей;
- поворот кода на заданный угол;
- выбор размера;
- возможность выбора готового шаблона.

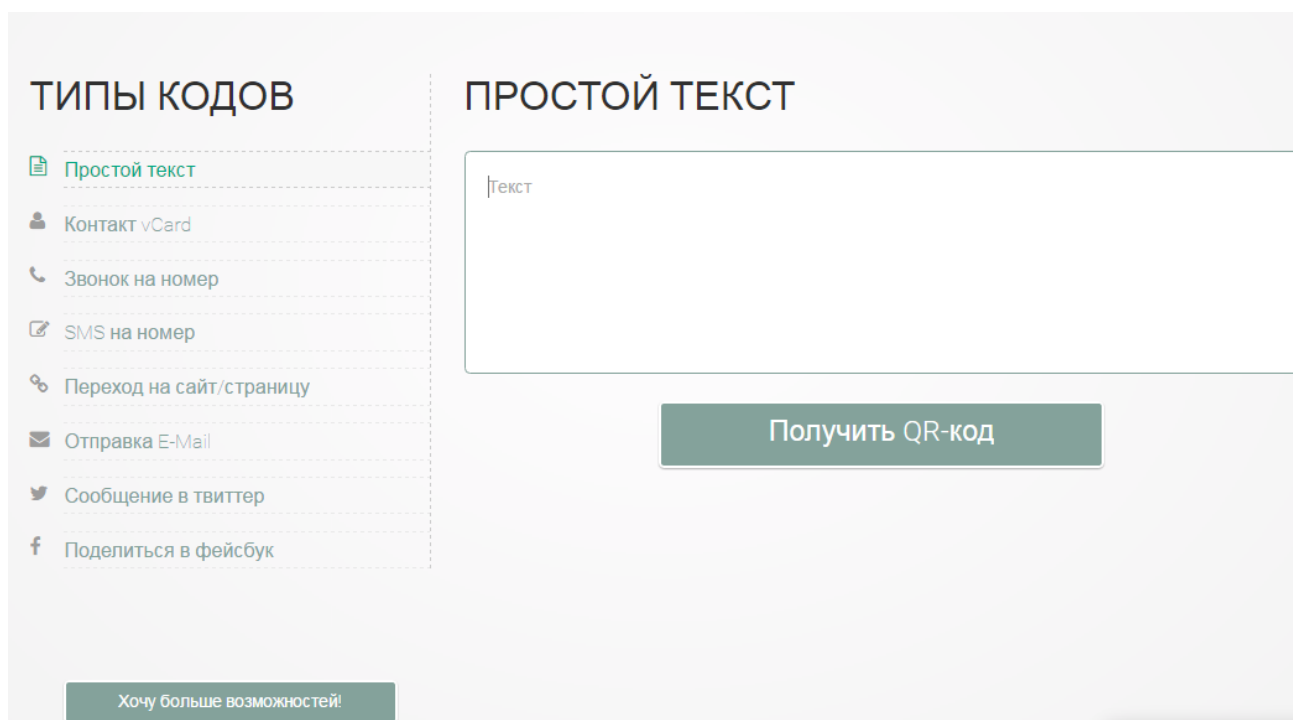


Рисунок 8. web-сервис creambee.ru

Qrcode.kaywa.com

Средствами этого сервиса можно сгенерировать QR-код содержащий следующую информацию [6]:

- адрес сайта; (бесплатно);
- SMS-сообщение;
- телефонный звонок;
- динамический QR-код.

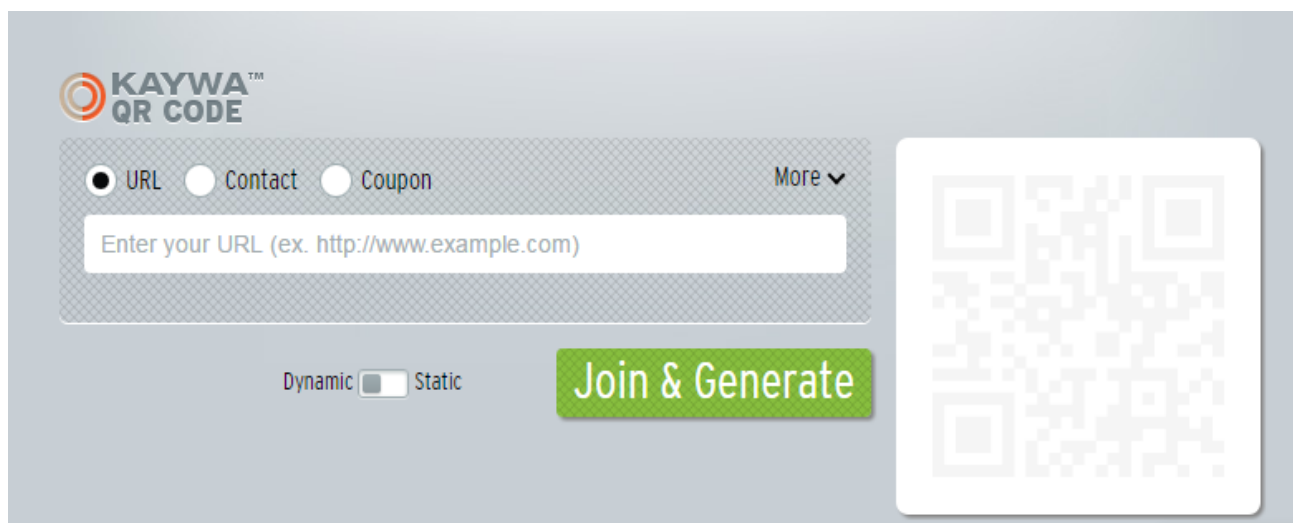


Рисунок 9. web-сервис qrcode.kaywa.com

Qr-code.com.ua

Этот сервис может сгенерировать QR-код, содержащий следующую информацию [5]:

- e-mail адрес;
- события календаря;
- геолокацию (координаты местонахождения объекта);
- URL-адрес;
- текст;
- SMS;
- номер телефона.

Имеется возможность задать дополнительные параметры:

- размер кода;
- уровень коррекции ошибок;
- кодировка.

Сохранить изображение с QR-кодом можно в формате PNG.

Тип кода: Контактная информация ▼

Имя

Компания

Название

Номер телефона

Email

Адрес

Адрес 2

Web-сайт

Пометка

Формат: MECARD ▼

Размер изображения: Большой ▼

Коррекция ошибок: L ▼

Кодировка: UTF-8 ▼

Генерировать →

Рисунок 10. web-сервис qr-code.com.ua

Ниже приведена таблица возможностей всех рассмотренных выше web-сервисов для генерации QR-кода (Таб.4).

Таблица 4.

*Возможности web-сервисов генерации QR-кодов*

Тип информации для кодирования	qrcoder.ru	qr-coder.ru	qrcc.ru	creambee.ru	qrcode.kaywa.com	qr-code.com.ua
Произвольный текст	+	+	+	+	+	+
Адрес сайта (URL)	+	+	+	+	+	+
Визитная карточка	+	+	+	+		+
SMS-сообщение	+	+	+	+	+	+
Телефонный звонок			+	+	+	+
e-mail адрес			+			+
e-mail сообщение			+	+		
Сообщение в Twitter		+		+		
Поделиться на Facebook				+		
Календарь событий			+			+
Wi-Fi сеть			+			+
Географическое положение (Google Maps)		+	+			+
Ссылка на YouTube		+				
Поиск на Android Market		+				
<b>Настраиваемые параметры QR-кода</b>						
Размер	+	+	+	+		+
Уровень коррекции ошибок		+				
Цвет			+			
Расширение выходного файла		+		+		
Текст рядом с кодом			+			
Логотип / Текст внутри QR-кода			+			
<b>Динамический QR-код</b>				+	+	

Таким образом, на основании проведенного сравнения можно сделать вывод о том, что web-сервис qrcc.com обладает рядом существенных преимуществ. Данный сервис предоставляет огромное количество возможностей для генерации QR-кода, а так же ряд настроек позволяющий настроить внешний вид QR-кода.

Динамические библиотеки (DLL):

- messagingtoolkit-qrcode;
- PHP QR Code;
- Kazuhiko Azare;
- QREncode;
- ZXing (Zebra Crossing).

messagingtoolkit-qrcode

Messagingtoolkit-qrcode является компонентом NET, который может быть использован для кодирования и декодирования QR-кода [14].

Библиотека представляет функции:

- кодировка QR-кода в изображения форматов JPEG, GIF, PNG или растровые форматы;
- декодирование QR-кода из изображения баркода.

Messagingtoolkit-qrcode разработана на объектно-ориентированном языке программирования C# и имеет открытый исходный код.

В отличие от других динамических библиотек (dll) работает быстрее.

Может использоваться в системах:

- WinForms приложений;
- Windows-приложений WPF;
- веб-приложений ASP.NET;
- Консольные приложения;
- Службы Windows;
- Windows Phone 7.x и 8.0;

- приложения для Windows 8;

Программу, построенную на библиотеке `messagingtoolkit-qrcode` можно настроить по параметрам:

- тип кодируемых данных;
- уровень коррекции ошибок;
- версия кода;
- размер модуля.

### PHP QR Code

Это библиотека с исходными кодами поддерживающая лицензию LGPL [3].

Предлагает набор программных средств для создания баркода в форматах PNG или JPEG. Полностью реализована на PHP.

Имеет немного настроек для создания баркода, а именно:

- Изменения уровня коррекции;
- Размер блока.

### Kazuhiko Azare

Японский программист реализовал алгоритм генерации QR-кода на различных языках программирования [1]:

- Java;
- ActionScript3;
- PHP;
- JavaScript.

Реализован только один тип кодирования: 8-битный.

Так же существует множество плагинов JQuery, предназначенных для генерации QR-кодов на сайте, в основном построенные на реализации JavaScript.

### QREncode

QREncode библиотека является свободным программным обеспечением и поддерживает лицензию открытого исходного кода LGPL [10].

У данной библиотеки существует большое количество реализаций, как на разных языках, так и для разных платформ, например, qtqrencode, реализованной на языке Qt и полностью оптимизированной для работы с этим языком.

#### ZXing (Zebra Crossing)

Zebra Crossing является лидером среди разработчиков в области QR-кодов [15].

Библиотека построена на разработке ZBar и реализована на объектно-ориентированном языке программирования Java.

Главным достоинством ZXing является то, что она имеет наименьший процент ошибок, в отличие от messagingtoolkit-barcode.

может использоваться в таких средах разработки как:

- Net 2.0, 3.5 и 4.0;
- Silverlight 4 и 5;
- Windows Phone 7.0, 7.1 and 8.0;
- Windows CE;
- Unity3D.

Исходя из представленного сравнения динамических библиотек для генерации QR-кода, можно сделать вывод о том, что самой функциональной библиотекой является messagingtoolkit-qrcode, в силу того, что она поддерживает большинство современных платформ разработки, а так же предоставляет ряд возможностей по настройке внешнего вида QR-кода.

Таким образом, в результате определения достоинств различных web-сервисов и сравнительного анализа динамических библиотек, можно говорить о том, что при создании удобного и функционального приложения для генерации QR-кодов целесообразно использовать динамическую библиотеку



messagingtoolkit-qrcode, а так же опираться на характеристики, функции и свойства web-сервиса qrcc.com

### **1.3 Техническое задание**

Составлено на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

#### **1. Общие сведения.**

##### **1.1. Название организации-заказчика.**

Уральский государственный педагогический университет  
(УрГПУ).

##### **1.2. Название продукта разработки (проектирования).**

Программа-приложение для генерации QR-кодов.

##### **1.3. Назначение продукта.**

Кодировка двумерных кодов по стандарту ГОСТ Р ИСО/МЭК 18004-2015, методом побитового кодирования знаков данных.

##### **1.4. Плановые сроки начала и окончания работ.**

- 1) сентябрь - составление содержательного плана ВКР;
- 2) сентябрь-октябрь - выбор объекта, обоснование актуальности темы, формулировка технического задания на разработку, выбор технологических решений и методов.
- 3) октябрь-февраль - критический анализ текущего состояния вопроса, выбор и обоснование путей решения, написание Главы 1.
- 4) октябрь-февраль - выполнение практической части исследования (разработка продукта, техдокументации, инструкций и т.п.).
- 5) февраль - промежуточная аттестация на заседании кафедры.
- 6) февраль-март - апробация разработанных материалов в учебном процессе.
- 7) март - подготовка публикации по теме исследования.

- 8) апрель-май - написание Главы 2 и заключительной части.
- 9) май-июнь - предварительная защита на заседании кафедры.  
окончательное оформление работы, подготовка доклада и презентации к защите.
- 10) первая неделя июня - Сдача ВКР в Государственную аттестационную комиссию.

## 2. Характеристика области применения продукта.

### 2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Продукт можно использовать для графического кодирования малого объема информации, и может применяться в любых сферах деятельности, например для реализации информационного обеспечения и управления учебным процессом.

### 2.2. Характеристика персонала (количество, квалификация, степень готовности)

Уверенный пользователь ПК.

## 3. Требования к продукту разработки.

### 3.1. Требования к продукту в целом.

Продукт представляет структуру классов, вместе которые преобразуют строку числовых или символьных данных в графический формат PNG.

### 3.2. Аппаратные требования.

Персональный компьютер.

### 3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

ОС: Любая удобная пользователю система семейства MS Windows.

### 3.4. Указание программного обеспечения, используемого для реализации.

Для реализации требуется поддержка оконного интерфейса.

3.5. Форматы входных и выходных данных

Программа на входе имеет некоторую строку информации, которая преобразуется в графический двумерный код и в последствии поступит на выход.

3.6. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Источником данных служит устройство ввода данных (клавиатура).

Вывод данных происходит непосредственно на экран, после чего вывод можно сохранить в виду bmp изображения или же скопировать в динамическую память компьютера (буфер обмена).

3.7. Порядок взаимодействия с другими системами, возможности обмена информацией.

Программа взаимодействует только с оперативной памятью компьютера, что позволяет ей обмениваться данными с любым приложением, принимающим графические данные.

3.8. Меры защиты информации.

Мер защиты безопасности данных не предусмотрено.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

Интуитивно понятный интерфейс.

4.2. Размещение информации на экране, дизайн экрана.

Для большего удобства, все управляющие компоненты размещены на одной форме. Ввод информации поступает в строку в верхней части окна. Управляющие кнопки находятся в нижней левой части окна. Вывод информации происходит в нижней правой части окна.

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

Техническое задание, Руководство пользователя.

## **Глава 2. Разработка приложения, генерирующего QR-код**

### **2.1 Информационные и функциональные модели приложения**

Перед тем как браться за разработку какого-либо приложения, важно знать, какие именно функции будет выполнять это приложение. Для того чтобы четко определить все функции будущей разработки, нужно построить функциональную модель.

Главной особенностью таких моделей является то, что с помощью блок-схем можно смоделировать приложение, определить все задачи, оптимизировать приложение и главное распределить временные рамки разработки приложения. В функциональных моделях отображаются все элементы проекта и связи между ними [17].

Функциональное моделирование использует графический язык описания процессов, поэтому любая функциональная модель будет выглядеть как совокупность упорядоченных диаграмм.

Для описания функциональных моделей в основном используются две методологии:

- диаграмма потоков данных (DFD);
- метод функционального моделирования (IDEF0).

В ходе исследования была выбрана динамическая библиотека для генерации QR-кодов `messagingtoolkit-qrcode`. Для описания функциональной модели алгоритма QR-кода использована диаграмма функционального моделирования IDEF0.

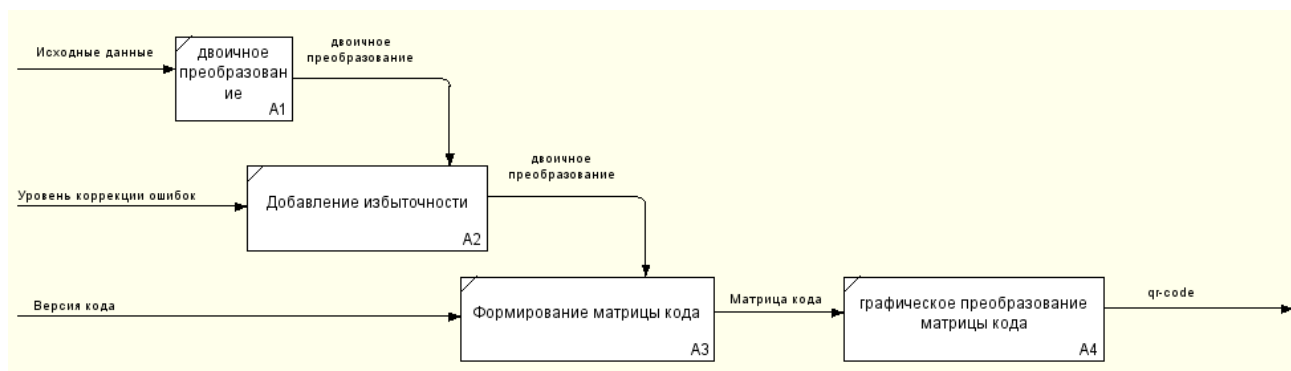


Рисунок 11. Функциональная модель библиотеки

На основе выбранной библиотеки необходимо разработать приложение для генерации QR-кодов. Ниже приведена структура разработанного приложения, для её моделирования была использована методология функционального моделирования IDEF0.

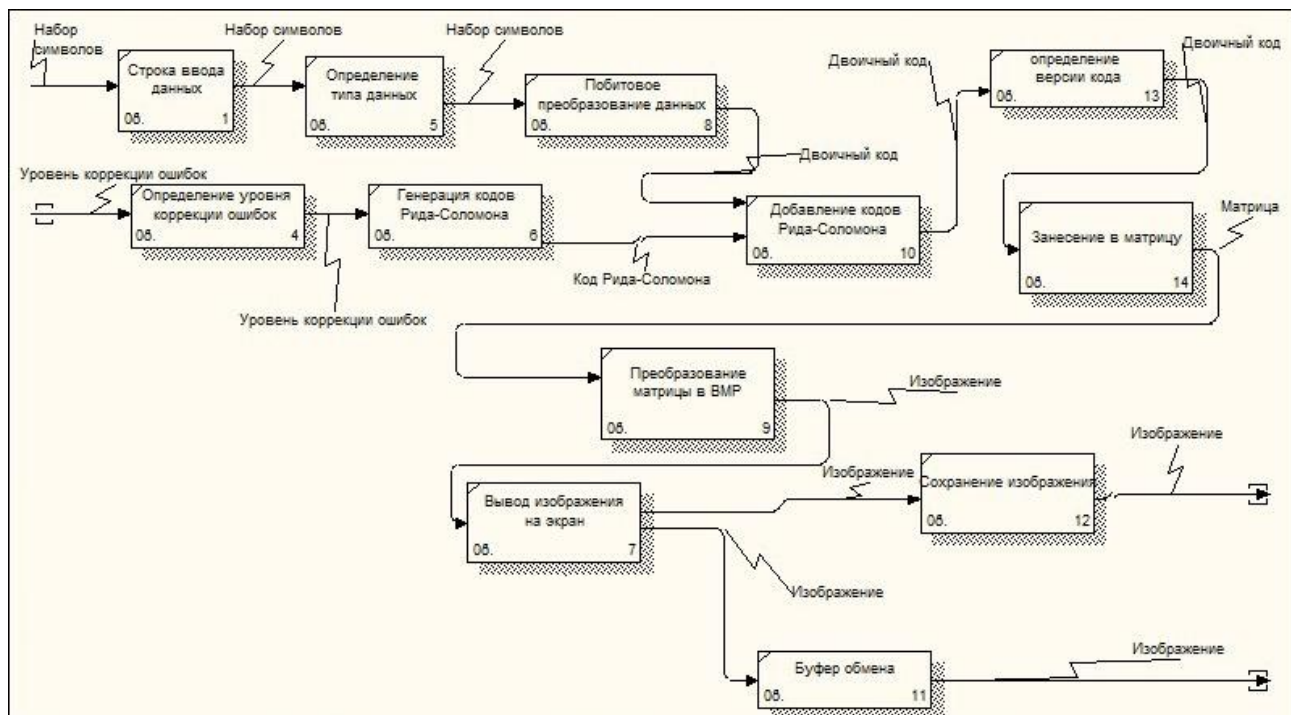


Рисунок 12. Функциональная модель приложения

Разработка приложения в среде программирования Visual Studio 2008:

Для приложения генерации QR-кода нам потребуется:

- Графический элемент для отображения QR-кода (pictureBox);
- Кнопки «Сгенерировать», «Сохранить», «Скопировать», «Загрузить», «Распознать» (button);

- Библиотека для генерации QR-кода (messagingtoolkit-qrcod).

Для начала нужно прикрепить библиотеку к проекту, это делается в обозревателе решений. Далее нужно ее инициализировать, для этого нужно перейти в редактор кода и в разделе «Using» пишем строки:

```
using MessagingToolkit.QRCode.Codec;  
using MessagingToolkit.QRCode.Codec.Data;
```

Далее перемещаем компоненты и расставляем их на свои места.

Для работы с динамической библиотекой будут использоваться команды:

- сгенерировать;
- декодировать.

Обработчик кнопки «Сгенерировать» выглядит следующим образом [18]:

*//если строка ввода данных не пуста*

```
if (textBox1.Text != "")
```

```
{ //инициализируем переменную qrtext
```

```
//и присваиваем ей строку ввода данных
```

```
string qrtext = textBox1.Text;
```

```
//создание объекта encoder класса QRCodeEncoder
```

```
QRCodeEncoder encoder = new QRCodeEncoder();
```

```
//кодируем qrtext в переменную qrcode класса Bitmap
```

```
Bitmap qrcode = encoder.Encode(qrtext);
```

```
//отправляем переменную qrcode в pictureBox1.
```

```
pictureBox1.Image = qrcode as Image;
```

```
}
```

```
else
```

```
{ //если строка ввода данных пуста то
```

```
//выводим системное сообщение об ошибке
```

```
MessageBox.Show("Возможно, вы не заполнили поле для  
кодирования.", "Ошибка");
```

```
}
```

А обработчик кнопки «Раскодировать» выглядит так [2]:

```
//если поле вывода QR-кода не пустое то
```

```
if (pictureBox1.Image != null)
```

```
{//Создаём объект decoder класса QRCodeDecoder
```

```
QRCodeDecoder decoder = new QRCodeDecoder();
```

```
//выводим системное сообщение с расшифрованным кодом
```

```
//в сообщении расшифровываем decoder из pictureBox
```

```
MessageBox.Show(decoder.decode(new
```

```
QRCodeBitmapImage(pictureBox1.Image as Bitmap )),( "Вывод"));
```

```
//отправляем дешифрованное сообщение в буфер обмена
```

```
Clipboard.SetData (DataFormats.Text, decoder.decode(new
```

```
QRCodeBitmapImage(pictureBox1.Image as Bitmap))));
```

```
}
```

```
else
```

```
{//если pictureBox пустой, то выводим сообщение об ошибке
```

```
MessageBox.Show("Возможно, нечего расшифровывать.",
```

```
"Ошибка");
```

```
}
```

Кнопки «Сохранить» и «Загрузить» используют стандартные диалоговые команды:

- SaveFileDialog
- OpenFileDialog

Кнопка «Скопировать код» имеет более сложную структуру [2]:

```
//находим pictureBox относительно экрана
```

```
Rectangle rect = pictureBox1.ClientRectangle;
```

```
//создаём изображение нужных размеров
```

```
Bitmap bmp = new Bitmap(rect.Width, rect.Height);
```



```

//перерисовываем содержимое pictureBox на Bitmap
pictureBox1.DrawToBitmap(bmp, rect);
//здаём размеры копируемой области
Rectangle copyRect = new Rectangle(0, 0, 180, 180);
//создаём изображение по размеру копируемой области
Bitmap bmpCopy = new Bitmap(copyRect.Width, copyRect.Height);
//получаем объект Graphics
using (Graphics g = Graphics.FromImage(bmpCopy))
//перерисовываем область с первоначальной картинки
g.DrawImage(bmp, 0, 0, copyRect, GraphicsUnit.Pixel);
//копируем в буфер обмена выделенную область
Clipboard.SetImage(bmpCopy);

```

## 2.2 Описание приложения (результата разработки)

В ходе исследования была поставлена задача разработать приложение для генерации QR-кодов на основе динамической библиотеки messagingtoolkit-qrcode. Приложение разрабатывалось на объектно-ориентированном языке программирования Visual C# в среде программирования Visual Studio 2008.

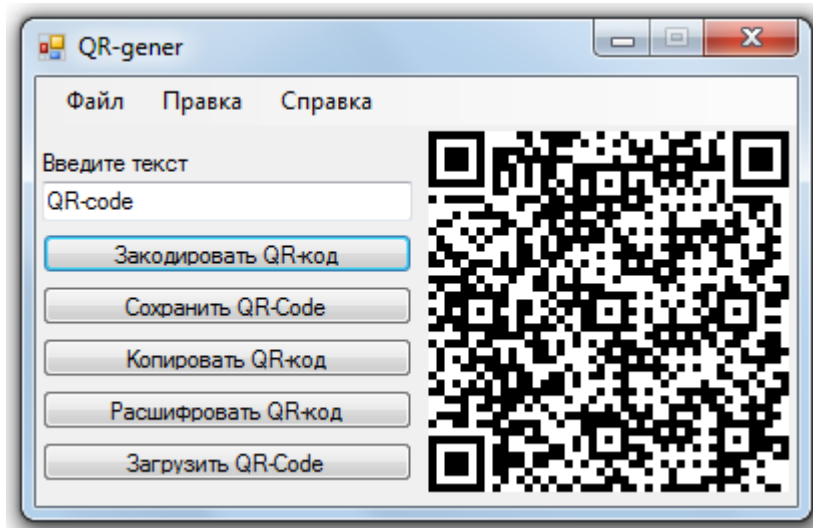


Рисунок 13. Приложение генерации QR-кода

Разработанное приложение поддерживает числовое и буквенно-числовое кодирование. Также при таком кодировании возможно закодировать URL-адрес, номер телефона, e-mail адрес, визитку. Приложение ставит наибольший уровень коррекции ошибок, что обеспечивает наибольшую безопасность данных в QR-коде. Так же приложение само определяет версию с помощью библиотеки.

В приложении есть возможность управления QR-кодом посредством:

- Создания QR-кода
- Сохранения QR-кода
- Копирование QR-кода
- Расшифровки QR-кода
- Загрузки QR-кода

Для генерации QR-кода достаточно ввести любое сообщение (на латинице или цифрами) в поле ввода текста и нажать «Закодировать QR-код». Если поле ввода информации пустое, то программа выведет сообщение об ошибке, а точнее, она говорит что поле сообщения пустое.

Для сохранения закодированного QR-кода необходимо нажать на кнопку «Сохранить QR-код». После нажатия на кнопку откроется диалоговое окно, где нужно будет выбрать место, куда сохранить сгенерированный QR-код и ввести имя для сохраняемого изображения QR-кода.

Если созданный QR-код необходимо скопировать в другое приложение, то можно нажать на кнопку «Скопировать QR-код», после чего перейти в другое приложение, куда нужно скопировать код и вставить скопированное изображения с помощью сочетания клавиш «Ctrl»+ «V» или с помощью специальной команды в приложении-получателе. QR-код не вставится в приложение, если приложение не принимает графическую информацию.

Если нужно раскодировать информацию из QR-кода, то сначала нужно нажать на кнопку «Загрузить QR-код», после чего откроется диалоговое окно

для выбора изображения QR-кода. Когда QR-код загрузится в приложение нужно нажать на кнопку «Расшифровать QR-код» и приложение выведет сообщение с раскодированной из QR-кода информацией и добавит скопированную информацию в буфер обмена, после чего эту информацию можно вставить в любое приложение, принимающее текстовую информацию (например web-обозреватель). Если поле вывода QR-кода пустое, то приложение выведет сообщение об ошибке.

Для большего удобства, команды с главной формы приложения продублированы в пункте «Файл».

Если нужно очистить поле ввода данных, то нужно нажать сперва кнопку «Правка», а затем кнопку «Очистить поле ввода».

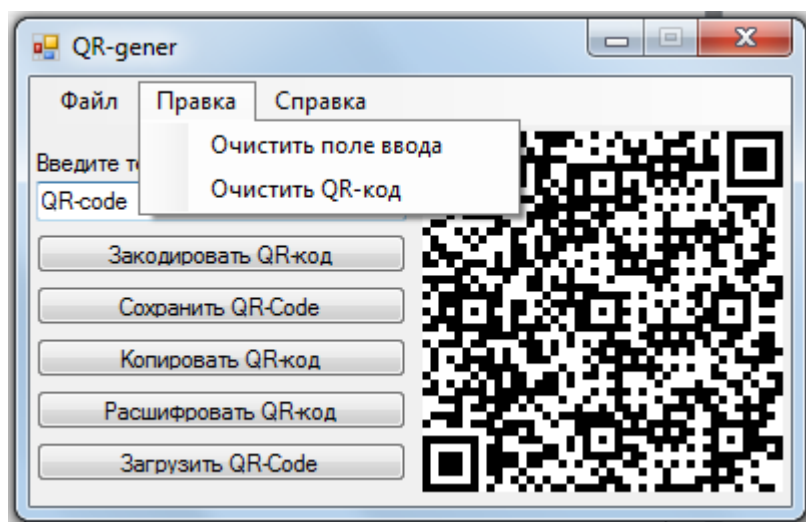


Рисунок 14. Приложение для генерации QR-кода

Аналогично можно очистить поле вывода QR-кода: «Правка» / «Очистить QR-код».

Так же для пользователей присутствует «Справка», где пользователь может посмотреть руководство пользователя и уточнить, как выполнить какую-либо задачу, например, загрузить картинку с QR-кодом в программу.

## 2.3 Апробация приложения

Для апробации разработанного приложения «Генератор QR-кодов» были приглашены студенты 4 курса, группы ИС-41, БИ-41 и Б-41. Во время апробации студентам было предоставлено разработанное приложение и анкета, которую они заполняли на основании анализа предложенной разработки. Анкета представляла собой систему экспертных оценок, разработанной в web-сервисе Google Forms. Студенты оценили приложение по критериям:

- удобство интерфейса
- эффективность приложения
- качество полученных результатов

Таким образом, в процессе апробации было опрошено 22 студента 4 курса, из них: 12 человек (54,54%) студенты группы ИС-41, 5 человек (22,73%) студенты группы БИ-41, 5 человек (22,73%) студенты группы Б-41.

В проведённом анкетировании были заданы вопросы:

1. Вы знаете что такое QR-код?

Вы знаете что такое QR-код? (22 ответа)

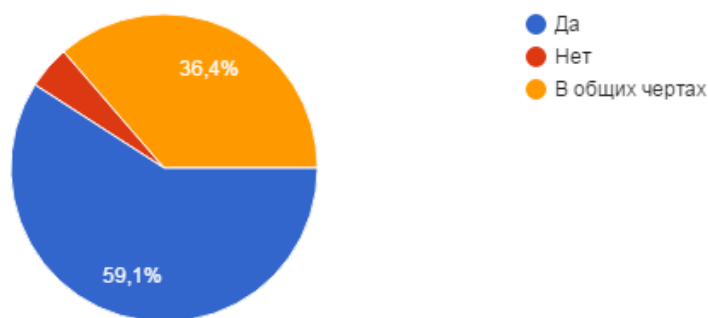


Рисунок 15. Диаграмма ответов на вопрос "Вы знаете что такое QR-код?"

## 2. Где вы раньше встречали QR-коды?

Где раньше вы встречали QR-коды? (22 ответа)

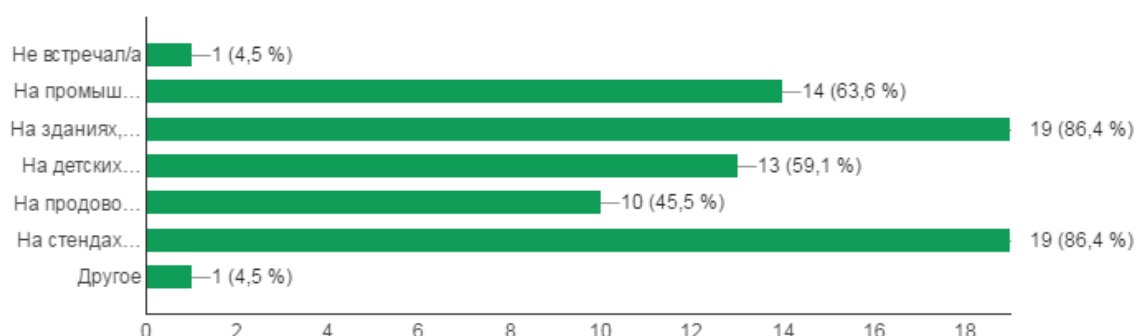


Рисунок 16. Диаграмма ответов на вопрос "Где вы раньше встречали QR-код?"

## 3. Пользовались ли Вы раньше приложениями генерации QR-кода?

Пользовались ли Вы раньше приложениями генерации QR-кодов? (22 ответа)

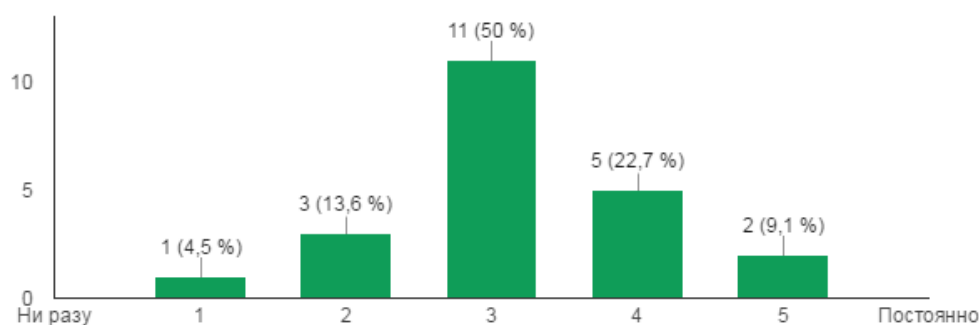


Рисунок 17. Диаграмма ответов на вопрос "Пользовались ли вы раньше приложениями генерации QR-кодов?"

4. Легко ли Вам было разобраться с тем, с работой данного приложения?
5. Понятен ли Вам интерфейс представленного приложения?
6. Понравилась ли Вам цветовая схема в которой выполнено данное приложение?
7. Данное приложение обладает всеми нужными функциями для приложений подобного рода?
8. Удобно ли Вам было работать с данным приложением?

9. Вызвало ли у Вас данное приложение желание узнать больше информации о технологии генерации QR-кодов?

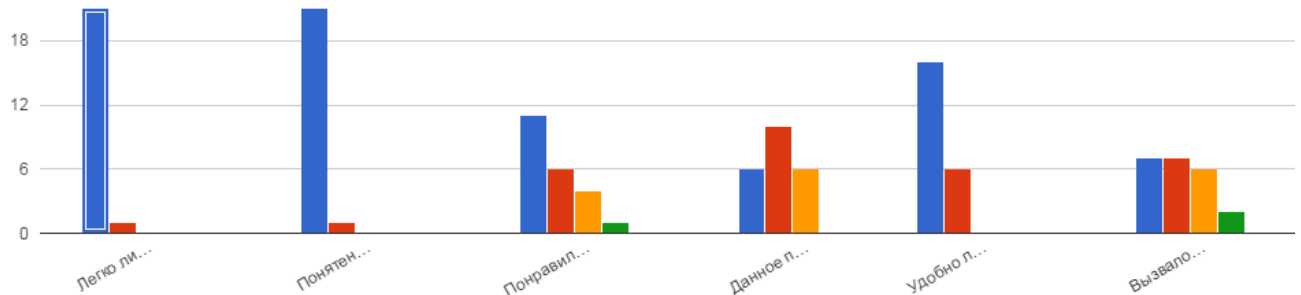


Рисунок 18. Диаграмма оценок характеристик приложения

По результатам анкетирования выяснилось, что большая часть респондентов знакома с QR-кодами и их применением, а так же пользуется QR-кодами.

По представленной статистике анкетирования, можно сделать вывод что, большинству студентов разработанный продукт был понятен и прост в использовании, а так же 64% респондентов заинтересовались технологиями генерации QR-кодов.

Основываясь на результатах данного анкетирования можно говорить о том, что данный продукт удобен, прост в использовании, имеет приятный глазу пользовательский интерфейс.

## **Заключение**

Данное исследование преследовало цель разработать приложение, реализующее генерацию QR-кода.

В проведённом исследовании была проанализирована научная и методическая литература в области QR-кодов. Из проведённого анализа литературы были сделаны выводы о структуре QR-кода, о его матрице и способах его кодирования. Матрица QR-кода, его структура, а так же все её составляющие части были подробно изучены.

В ходе исследования были рассмотрены наиболее популярные web-сервисы для генерации QR-кода и произведено сравнение возможностей этих сервисов. На основе этого сравнения был выявлен самый функциональный web-сервис – qrcc.com. Функциональные возможности данного сервиса намного выше чем у остальных подобных сервисов.

Так же были проанализированы динамические библиотеки для создания приложений генерации QR-кодов. В результате анализа библиотек была выбрана библиотека messagingtoolkit-qrcode для дальнейшего построения приложения-генератора QR-кодов на основе этой библиотеки, так как она обладает наиболее мощной работоспособностью и наиболее большим функционалом, чем у остальных рассматриваемых библиотек генерации QR-кода.

Для построения приложения-генератора QR-кодов было разработано техническое задание, на основании которого и проводились дальнейшие работы по созданию приложения. Основной упор в разработке приложения был на генерацию QR-кода.

На основе выбранной динамической библиотеки было построено приложение, позволяющее генерировать QR-код и распознавать его с загруженной картинки. В приложении реализовано два способа вывода сгенерированного QR-кода: сохранить как картинку и скопировать в буфер

обмена. Разработанное приложение разрабатывалось в среде Visual Studio 2008. Среда разработки была выбрана в силу универсальности разрабатываемых на ней приложений. В качестве языка программирования, на котором было разработано приложение генерации QR-кода, был выбран Visual C#, благодаря тому, что он набирает всё большую популярность среди разработчиков.

В конце исследования была проведена апробация разработанного приложения, в ходе которой были выявлены плюсы и минусы разработанного приложения для генерации QR-кодов а так же составлена статистика осведомлённости студентов о технологии QR-кодов и их использования.

В результате последовательного выполнения выявленных задач была достигнута поставленная цель разработать приложение, реализующее генерацию QR-кодов.



## Литература

1. Kazuhiko Azare URL: <https://code.google.com/archive/p/luajoint/>
2. msdn.microsoft.com URL: [https://msdn.microsoft.com/ru-ru/library/system.windows.clipboard\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.windows.clipboard(v=vs.110).aspx)
3. PHP QR Code URL: <http://phpqrcode.sourceforge.net>
4. qrcc.ru URL: <http://qrcc.ru/generator.php>
5. qrcode.com.ua URL: <http://qrcode.com.ua>
6. qrcode.kaywa.com URL: <http://qrcode.kaywa.com>
7. qrcoder.ru URL: <http://qrcoder.ru>
8. qr-coder.ru URL: <http://qr-coder.ru>
9. qreambee.ru URL: <http://qreambee.ru>
10. QREncode URL: <https://packages.altlinux.org/ru/Sisyphus/srpms/qrencode>
11. QR-код. Добавление служебной информации // wikipedia.com URL: <https://ru.wikipedia.org/wiki/QR-код>
12. QR-код. Этап размещения информации на поле кода // wikipedia.com URL: <https://ru.wikipedia.org/wiki/QR-код>
13. QR-код. Кодирование // wikipedia.com URL: <https://ru.wikipedia.org/wiki/QR-код>
14. twit88.com - messaging made easy URL: <http://platform.twit88.com>
15. ZXing (Zebra Crossing) URL: <https://github.com/zxing/zxing/wiki/Getting-Started-Developing>
16. ГОСТ Р ИСО/МЭК 18004-2015 Спецификация символики штрихового кода QR-code / Москва: Стандартинформ, 2015. 113 с.
17. МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ IDEF0 / Москва: ГОССТАНДАРТ РОССИИ, 2000. 75 с.

- 18.О.М. Котов Язык С#. Краткое описание и введение в технологии программирования. Екатеринбург: Издательство Уральского университета, 2014. 208 с.
- 19.Таблица Unicode // Хитрые инструменты URL:  
<http://foxtools.ru/Unicode>
- 20.Читаем QR код // geektimes.ru URL: <https://geektimes.ru/post/256932/>

## Приложения

### Приложение 1. Листинг программы

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MessagingToolkit.QRCode.Codec;
using MessagingToolkit.QRCode.Codec.Data;

namespace QR_Cod
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        String result = System.String.Empty;

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "")
            {
                string qrtext = textBox1.Text;
                //считываем текст из TextBox'a
                QRCodeEncoder encoder = new QRCodeEncoder();
                //создаем объект класса QRCodeEncoder
                Bitmap qrcode = encoder.Encode(qrtext);
            }
        }
    }
}
```

```

        /* кодируем слово, полученное из TextBox'a (qrcode) в
        переменную qrcode. класса Bitmap(класс, который используется для
        работы с изображениями)*/
        pictureBox1.Image = qrcode as Image;
        // pictureBox выводит qrcode как изображение.
    }
    else
    {
        MessageBox.Show("Возможно, вы не заполнили поле для
        кодирования.", "Ошибка");
    }
}

private void button2_Click(object sender, EventArgs e)
{
    SaveFileDialog save = new SaveFileDialog();
    /*save будет запрашивать у пользователя, место, в которое он
    захочет сохранить файл. */
    save.Filter = "PNG|*.png|JPEG|*.jpg|GIF|*.gif|BMP|*.bmp";
    /*создаём фильтр, который определяет, в каких форматах мы
    сможем сохранить нашу информацию. В данном случае выбираем
    форматы изображений. Записывается так: "название_формата_в
    обозревателе|*.расширение_формата")*/
    if (save.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    //если пользователь нажимает в обозревателе кнопку "Сохранить"
    {
        pictureBox1.Image.Save(save.FileName);
        /*изображение из pictureBox'a сохраняется под именем,
        которое введёт пользователь*/
    }
}

```

```

    }
}

private void button3_Click(object sender, EventArgs e)
{
    OpenFileDialog load = new OpenFileDialog();
    /*load будет запрашивать у пользователя место, из которого он
хочет загрузить файл.*/
    if (load.ShowDialog() == System.Windows.Forms.DialogResult.OK)
        //если пользователь нажимает в обозревателе кнопку "Открыть".
        {
            pictureBox1.ImageLocation = load.FileName;
            /*в pictureBox'e открывается файл, который был по пути,
запрошенном пользователем.*/
        }
}

private void button4_Click(object sender, EventArgs e)
{
    if (pictureBox1.Image != null)
    {
        QRCodeDecoder decoder = new QRCodeDecoder();
        // создаём "раскодирование изображения"
        MessageBox.Show(decoder.decode(new
QRCodeBitmapImage(pictureBox1.Image as Bitmap)));
        /*в MessageBox'e программа запишет раскодированное
сообщение с изображения, которое предоврительно будет
переведено из pictureBox'a в класс Bitmap, чтобы мы смогли с этим
изображением работать.*/
    }
}

```

```

        Clipboard.SetData (DataFormats.Text, decoder.decode(new
QRCodeBitmapImage(pictureBox1.Image as Bitmap)));
    }
else
{
    MessageBox.Show("Возможно, нечего расшифровывать.",
"Ошибка");
}
}

private void button5_Click(object sender, EventArgs e)
{
    // Область PictureBox
    Rectangle rect = pictureBox1.ClientRectangle;
    // Создаём изображение нужных размеров
    Bitmap bmp = new Bitmap(rect.Width, rect.Height);
    // Рисуем содержимое PictureBox на Bitmap
    pictureBox1.DrawToBitmap(bmp, rect);
    // Задаём размеры копируемой области
    Rectangle copyRect = new Rectangle(0, 0, 180, 180);
    // Создаём изображение по размеру копируемой области
    Bitmap bmpCopy = new Bitmap(copyRect.Width, copyRect.Height);
    // Получаем объект Graphics
    using (Graphics g = Graphics.FromImage(bmpCopy))
    // Перерисовываем область с первоначальной картинки
    // bmp - откуда копируем
    // 0, 0 - координаты левого верхнего угла на новой картинке
    // copyRect - область из которой срисовываем
    // Единицы измерения для перерисовки

```

```

        g.DrawImage bmp, 0, 0, copyRect, GraphicsUnit.Pixel);
        // Кладём в буфер обмена "вырезанный" кусок
        Clipboard.SetImage bmpCopy);
    }
    private void очиститьПолеВводаToolStripMenuItem_Click(object sender,
        EventArgs e)
    {
        textBox1.Text = "";
    }
    private void очиститьQRкодToolStripMenuItem_Click(object sender,
        EventArgs e)
    {
        pictureBox1.Image = null;
    }
}
}

```